

2. Aufgabe: Grundlagen

Schreiben Sie ein vollständiges und lauffähiges C-Hauptprogramm, das die Kantenlängen eines Rechtecks als reelle Zahlen einliest und dessen Fläche berechnet. Die Fläche soll kaufmännisch gerundet werden und als ganze Zahl vorliegen. Negative Werte brauchen nicht berücksichtigt bzw. abgefangen werden. Anschließend soll der gerundete Umfang auf die Konsole ausgegeben werden. Berücksichtigen Sie bei der Ausgabe auf die Konsole eine passende Beschriftung. (15 P)

3. Aufgabe: Funktionen

a) Erklären Sie aus welchen Teilen eine Funktion besteht (strukturiert vorgehen!) (14 P):

b) Wie bezeichnet man bei folgenden Beispielen die Art, mit der die Parameter der Funktion übergeben werden? Erklären Sie den Mechanismus in **kurzen Stichworten** (6 P):

i. `int function1(double* pdValue)`

ii. `void function2(int iValue)`

4. Aufgabe: Array/Feld, Indizierung

a) Erklären Sie den Unterschied zwischen einem Array und einer Struktur (kein Roman – kann Punktabzug geben!). (4 P)

b) Schreiben Sie eine kleine Funktion, bei der ein Array mit reellen Messwerten übergeben wird und der Mittelwert bestimmt wird. Der Mittelwert soll als reelle Zahl in Form eines Funktionswertes zurückgegeben werden. Geben Sie anschließend ein Programmfragment an, bei dem die von Ihnen definierte Funktion aufgerufen wird (d.h. weniger als 5 Anweisungen reichen und bitte kein Hauptprogramm oder irgendwelche Ein- oder Ausgaben schreiben!). Bibliotheken dürfen nicht verwendet werden! (10 P)

5. Aufgabe: Zeichenketten

a) Schreiben Sie eine Funktion `strlen`, die die Länge einer mit 0 terminierten Zeichenkette bestimmt. Die Zeichenkette soll als Parameter übergeben werden. Der Funktionswert soll die Länge zurückgeben (kein Hauptprogramm, keine Ein- oder Ausgabe!). (10 P)

b) Schreiben Sie eine Funktion `count`, die die Anzahl von Zeichen, die KEINE Großbuchstaben sind in einer Zeichenkette `acString` zählt. (kein Hauptprogramm, keine Ein- oder Ausgabe!). (14 P)

`count("This IS An ExamPle!") → 13`

Hinweis: Denken Sie daran, dass die Zeichen 'A' ... 'Z' wie Zahlen benutzt werden können.

6. Aufgabe: Algorithmus

Was machen die nachfolgenden Funktionen unbekannt1-4?

Bitte beschreiben Sie die Funktionsweise möglichst abstrakt – Romane geben Abzug! (15 P)

Hinweis: Testen Sie den Algorithmus anhand eines Funktionsaufrufs und beobachten Sie die Variablenwerte.

```
#include <stdlib.h>

int unbekannt1(int iZahl1, int iZahl2)
{
    int iZahl3 = 1;
    for (int i = 0; i < iZahl2; ++i)
    {
        iZahl3 *= iZahl1;
    }
    return iZahl3;
}

int unbekannt2(double dZahl)
{
    return (int)(dZahl + (dZahl > 0.0? 0.5 : -0.5));
}

double unbekannt3(double dZahl1, int iZahl2)
{
    int iZahl3 = unbekannt1(10, iZahl2);
    return unbekannt2(dZahl1 * iZahl3) / ((double)iZahl3);
}

double* unbekannt4(int iZahl1)
{
    double* adValues = (double*)malloc(iZahl1 * sizeof(double));
    for (int i = 0; i < iZahl1; ++i)
    {
        adValues[i] = unbekannt3(i * 3.1415926539, 3);
    }
    return adValues;
}
```